

FIG. 1
(PRIOR ART)

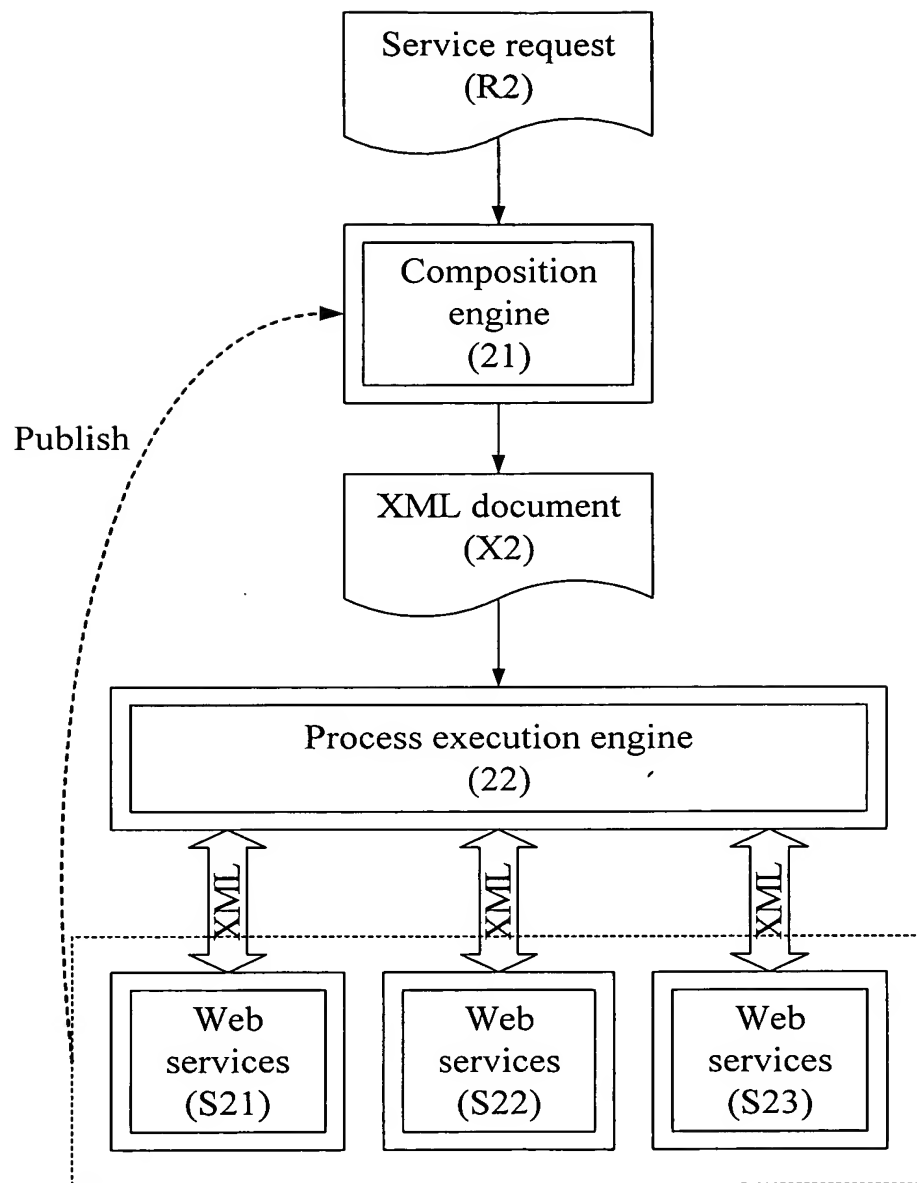


FIG. 2
(PRIOR ART)

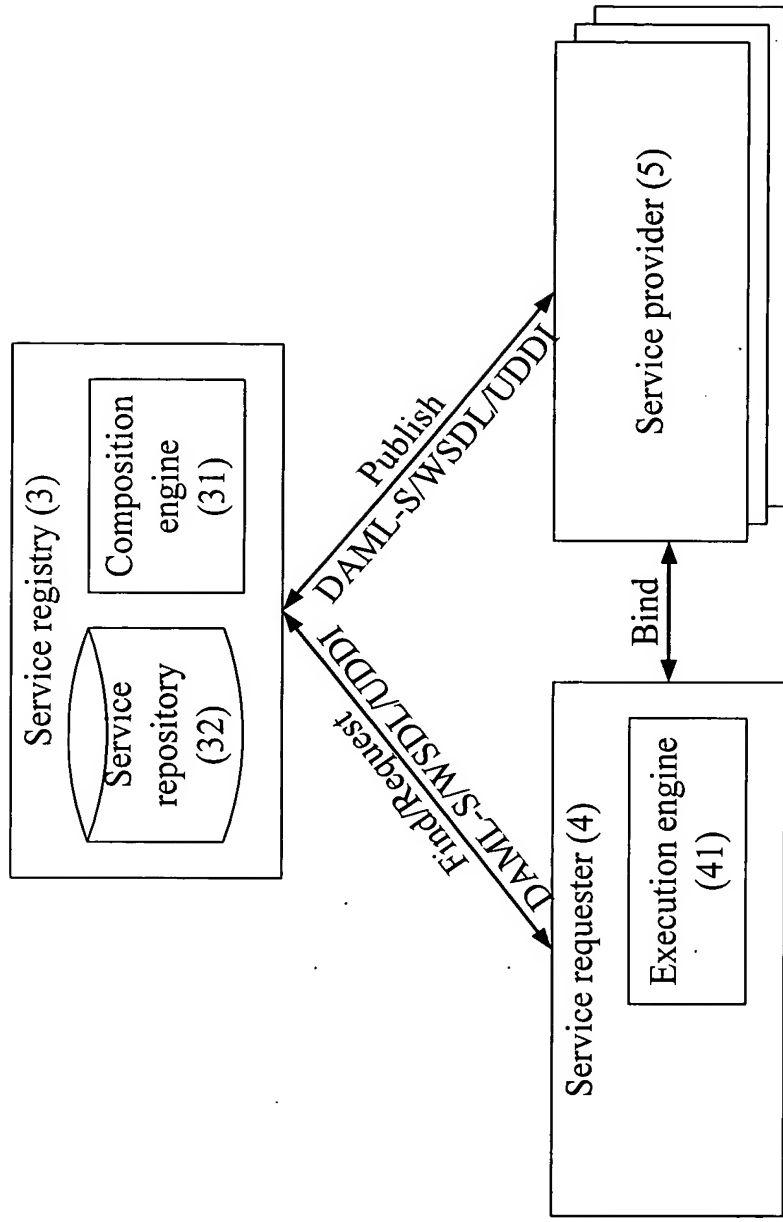


FIG. 3

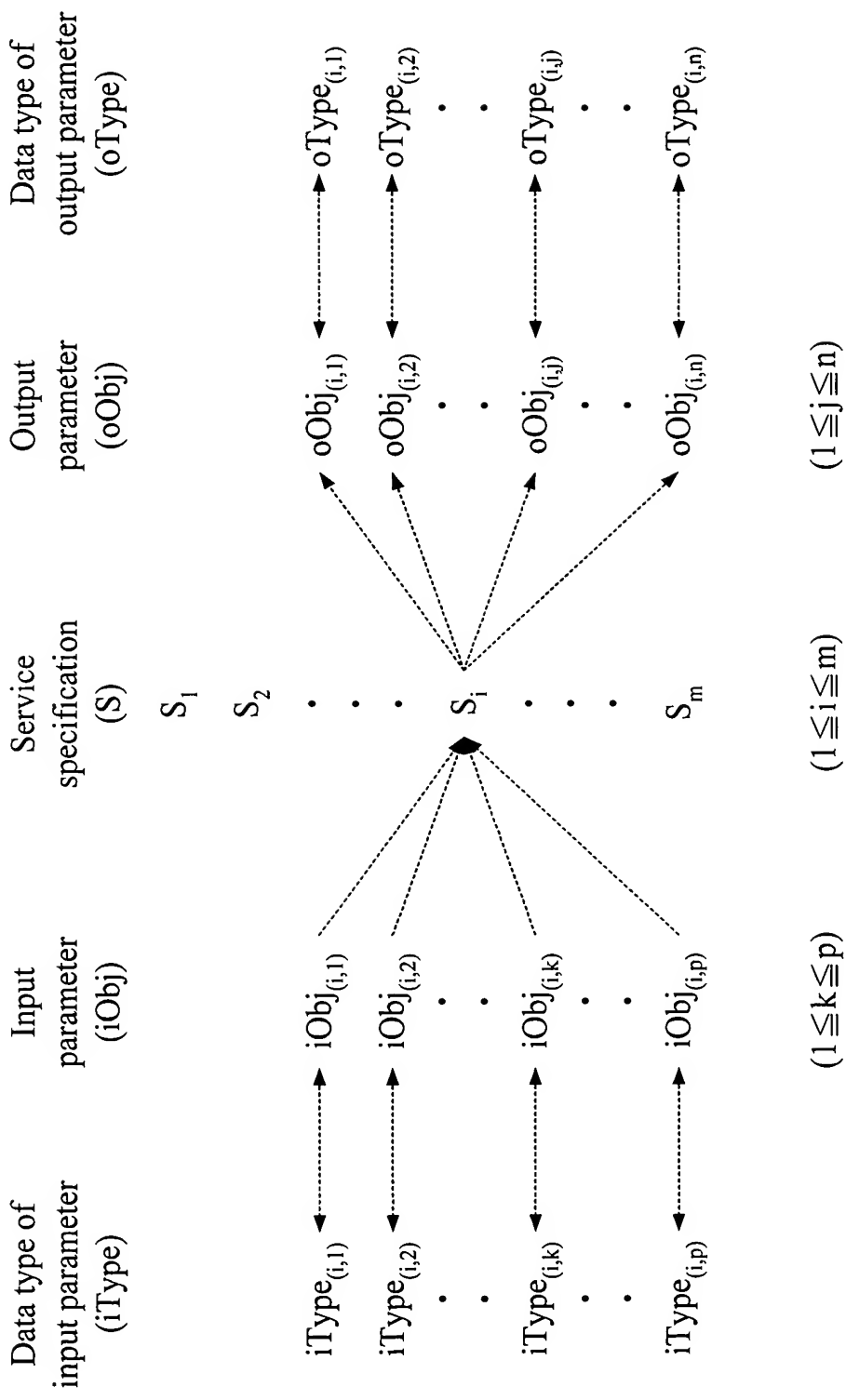


FIG. 4

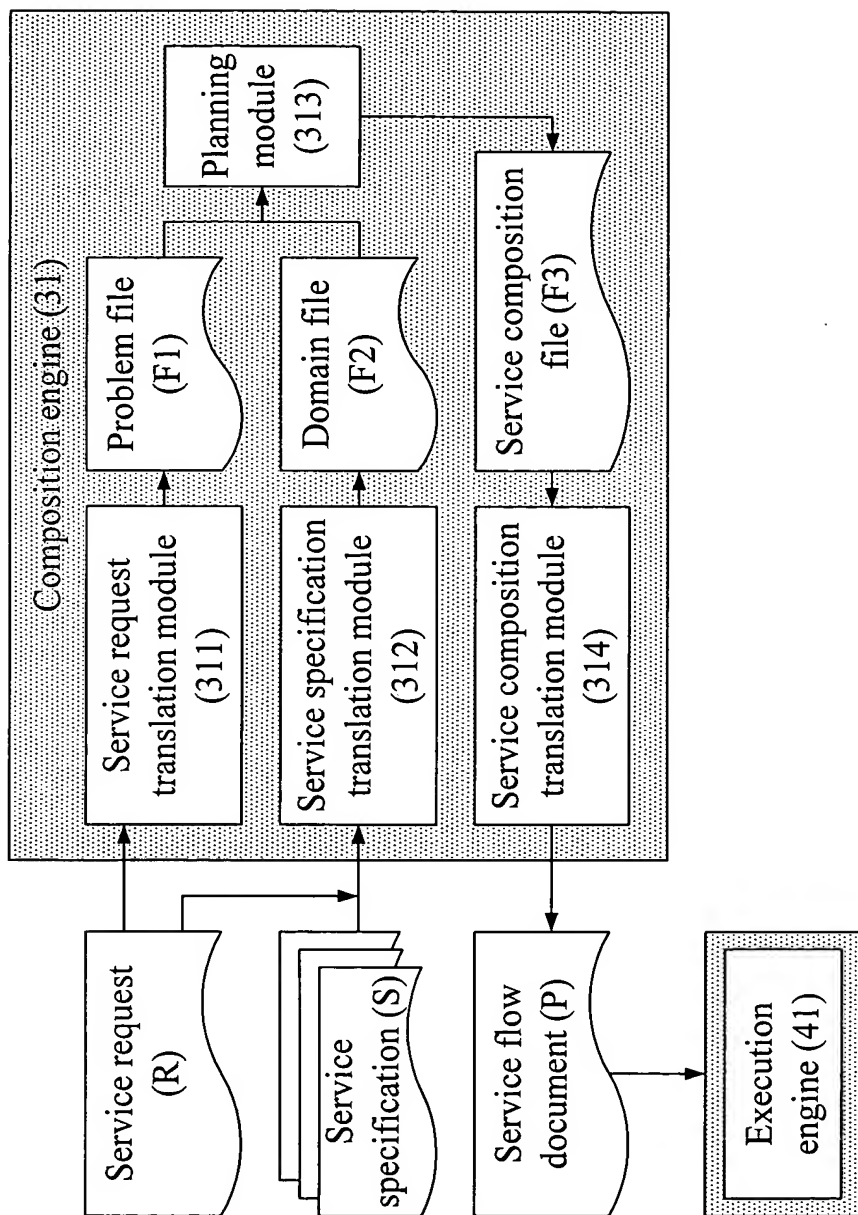


FIG. 5

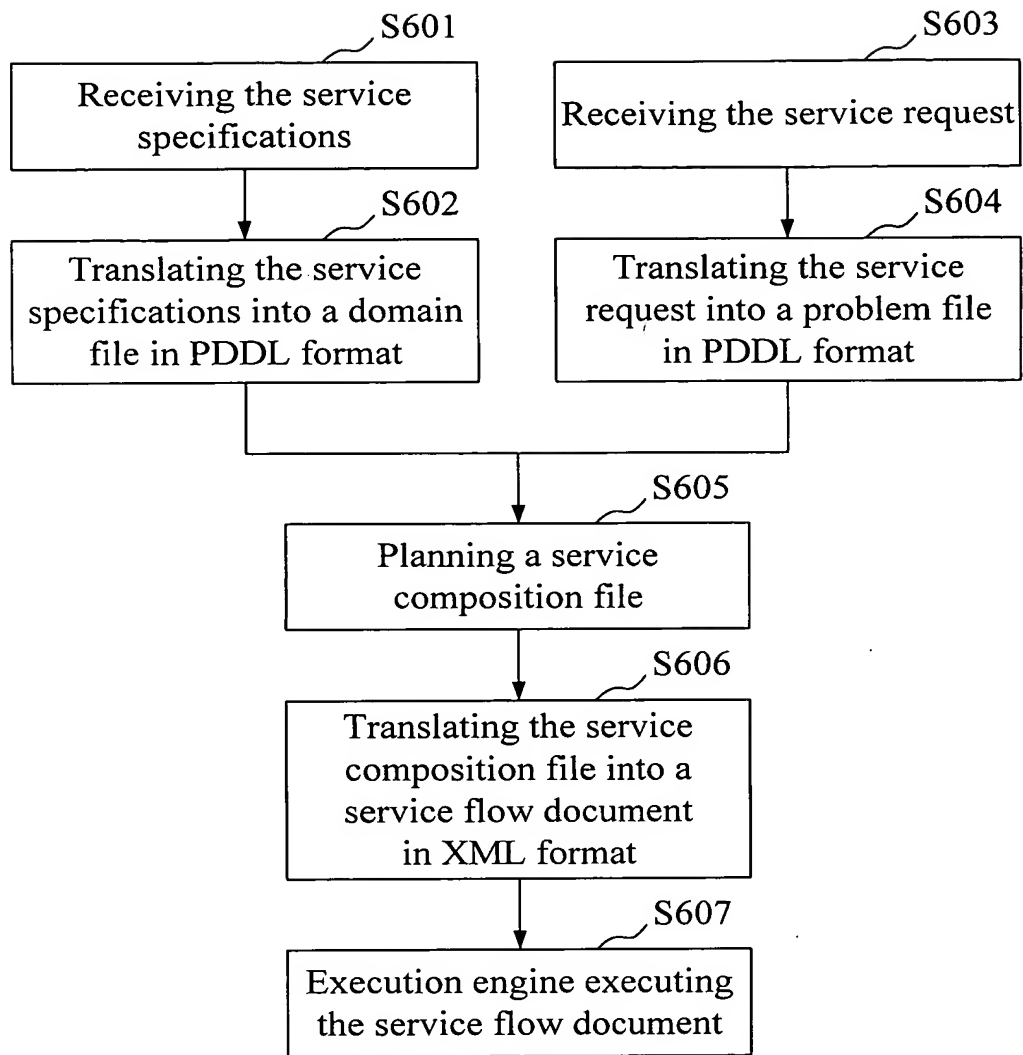


FIG. 6

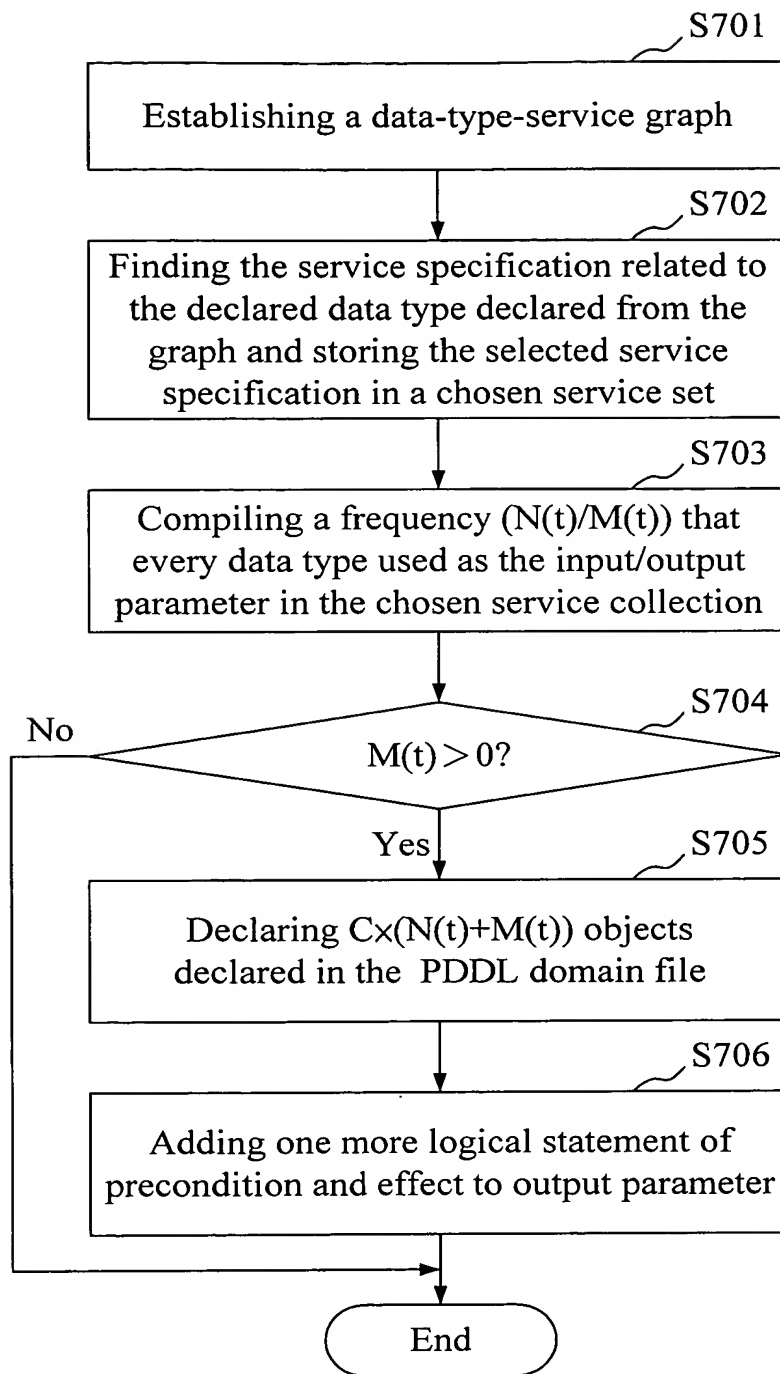


FIG. 7

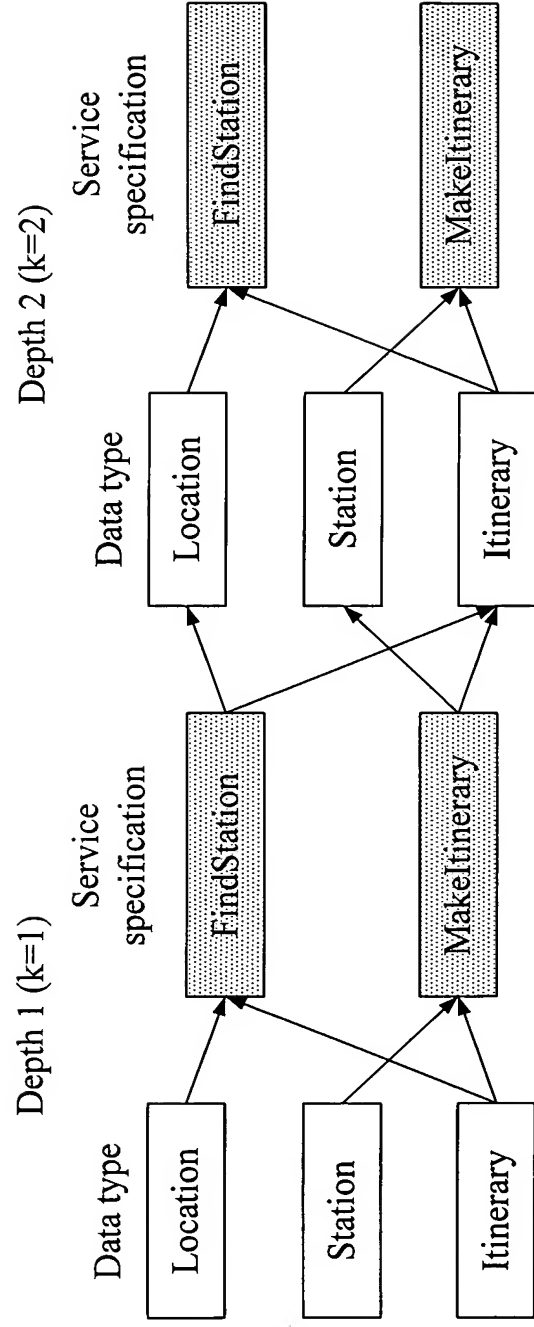


FIG. 8

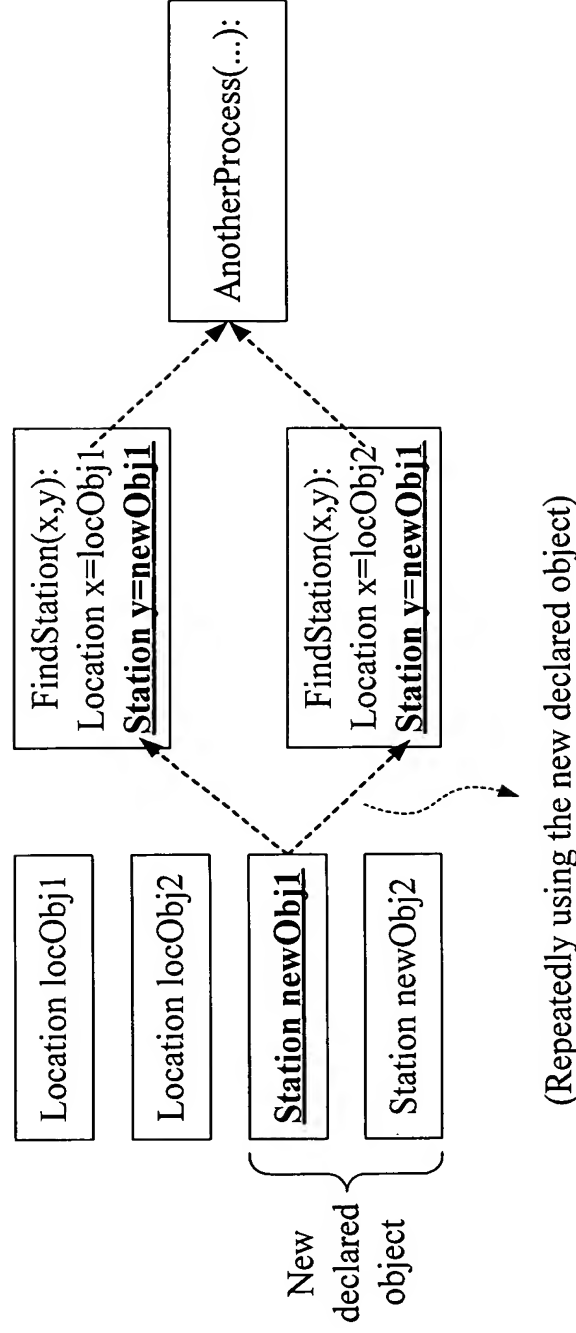


FIG. 9

PDDL problem file program code

```
(define (problem iTour1)
  (:domain iTour)
  (:objects locObj1 locObj2 - Location
            itineraryObj - Itinerary)

  (:init (Start    locObj1)
         (End      locObj2))
  (:goal (and (Trip itineraryObj locObj1 locObj2))))
```

FIG. 10

PDDL domain file program code

```
(define (domain iTour)
  (:requirements :strips :equality :typing)
  (:predicates
    (Start ?loc - Location)
    (End ?loc - Location)
    (At ?station - Station
      ?loc - Location)
    (Trip ?i - Itinerary
      ?loc1 ?loc2 - Location)
  )

  (:action AskStation
    :parameters (?loc - Location
      ?station - Station)
    :effect (and (At ?station ?loc))
  )

  (:action MakeItinerary
    :parameters (?loc1 ?loc2 - Location
      ?station1 ?station2 - Station
      ?i - Itinerary)
    :precondition (and (Start ?loc1)
      (End ?loc2)
      (At ?station1 ?loc1)
      (At ?station2 ?loc2))
    :effect (and (Trip ?i ?loc1 ?loc2))
  )
)
```

FIG. 11

PDDL domain file program code

```

(define (domain iTour)
  (:requirements :strips :equality :typing)
  (:predicates
    (Start ?loc - Location)
    (End ?loc - Location)
    (At ?station - Station
      ?loc - Location)
    (Trip ?i - Itinerary
      ?loc1 ?loc2 - Location)
    (used ?o - Object)
  )

  (:constants newStationObj1 newStationObj2 newStationObj3 - Station
    newItineraryObj - Itinerary)

  (:action AskStation
    :parameters (?loc - Location
      ?station - Station)
    :precondition (and (not (used ?station)) )
    :effect (and (At ?station ?loc)
      (used ?station)
    )
  )

  (:action MakeItinerary
    :parameters (?loc1 ?loc2 - Location
      ?station1 ?station2 - Station
      ?i - Itinerary)
    :precondition (and (Start ?loc1)
      (End ?loc2)
      (At ?station1 ?loc1)
      (At ?station2 ?loc2)
      (not (used ?i))
    )
    :effect (
      and (Trip ?i ?loc1 ?loc2)
      (used ?i)
    )
  )
)

```

FIG. 12